

# Performance Evaluation of a Terrain Traversability Learning Algorithm in The DARPA LAGR Program

Michael Shneier, Will Shackelford, Tsai Hong and Tommy Chang  
Intelligent Systems Division  
National Institute of Standards and Technology  
Gaithersburg, MD 20899  
{firstname.lastname}@nist.gov

**Abstract**—The Defense Applied Research Projects Agency (DARPA) Learning Applied to Ground Vehicles (LAGR) program aims to develop algorithms for autonomous vehicle navigation that learn how to operate in complex terrain. For the LAGR program, The National Institute of Standards and Technology (NIST) has embedded learning into a control system architecture called 4D/RCS to enable the small robot used in the program to learn to navigate through a range of terrain types. This paper describes performance evaluation experiments on one of the algorithms developed under the program to learn terrain traversability. The algorithm uses color and texture to build models describing regions of terrain seen by the vehicle's stereo cameras. Range measurements from stereo are used to assign traversability measures to the regions. The assumption is made that regions that look alike have similar traversability. Thus, regions that match one of the models inherit the traversability stored in the model. This allows all areas of images seen by the vehicle to be classified, and enables a path planner to determine a traversable path to the goal.

The algorithm is evaluated by comparison with ground truth generated by a human observer. A graphical user interface (GUI) was developed that displays an image and randomly generates a point to be classified. The human assigns a traversability label to the point, and the learning algorithm associates its own label with the point. When a large number of such points have been labeled across a sequence of images, the performance of the learning algorithm is determined in terms of error rates. The learning algorithm is outlined in the paper, and results of performance evaluation are described.

**Keywords:** Learning, performance evaluation, traversability, computer vision, robotics, LAGR

## I. INTRODUCTION

The Defense Applied Research Projects Agency (DARPA) Learning Applied to Ground Vehicles (LAGR) program [1] aims to develop algorithms for autonomous vehicle navigation that learn how to operate in complex terrain. Over many years, the National Institute of Standards and Technology (NIST) has developed a reference model control system architecture called 4D/RCS that has been applied to many kinds of robot control, including autonomous vehicle control [2]. For the LAGR program, NIST has embedded learning into a 4D/RCS

controller to enable the small robot used in the program to learn to navigate through a range of terrain types [3]. The vehicle learns in several ways. These include learning by example, learning by experience, and learning how to optimize traversal. In this paper, we present a method of evaluating a learning algorithm used in LAGR that associates terrain appearance with traversability. The paper briefly describes the learning method and then focuses on the evaluation procedure. The approach is illustrated with examples taken from tests run by the LAGR evaluation team.

The appearance of regions in an image has been described in many ways, but most frequently in terms of color and/or texture. Ulrich and Nourbakhsh [4] used color imagery to learn the appearance of a set of locations to enable a robot to recognize where it is. A set of images was recorded at each location and served as descriptors for that location. Images were represented by a set of one-dimensional histograms in both HLS (hue, luminance, saturation) and normalized Red, Green, and Blue (RGB) color spaces. When the robot needed to recognize its location, it compared its current image with the set of images associated with locations. The location was recognized as that associated with the best-matching stored image.

In [5] the authors also addressed the issue of appearance-based obstacle detection using a single color camera and no range information. Their approach makes the assumptions that the ground is flat and that the region directly in front of the robot is ground. This region is characterized by color histograms and used as a model for ground. In the domain of road detection, a related approach is described in [6]. In principle, the method could be extended to deal with more classes, and our algorithm can be seen as one such extension that does not need to make the assumptions because of the availability of range information for regions close to the vehicle.

Learning has been applied to computer vision for a variety of applications, including traversability prediction. Wellington and Stentz [7] predicted the load-bearing surface under vegetation by extracting features from range data and associating them with the actual surface height measured

Report Documentation Page				Form Approved OMB No. 0704-0188	
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.					
1. REPORT DATE <b>2009</b>		2. REPORT TYPE		3. DATES COVERED <b>00-00-2009 to 00-00-2009</b>	
4. TITLE AND SUBTITLE <b>Performance Evaluation Of A Terrain Traversability Learning Algorithm In The DARPA LAGR Program</b>				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S)				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) <b>Intelligent Systems Division, National Institute of Standards and Technology, Gaithersburg, MD, 20899</b>				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT <b>Approved for public release; distribution unlimited</b>					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT <b>See Report</b>					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT <b>Same as Report (SAR)</b>	18. NUMBER OF PAGES <b>8</b>	19a. NAME OF RESPONSIBLE PERSON
a. REPORT <b>unclassified</b>	b. ABSTRACT <b>unclassified</b>	c. THIS PAGE <b>unclassified</b>			

when the vehicle drove over the corresponding terrain. The system learned a mapping from terrain features to surface height using a technique called locally weighted regression. Learning was done in a map domain. We also use a map in the current work, although it is a two dimensional (2D) rather than a three dimensional (3D) map, and we also make use of the information gained when driving over terrain to update traversability estimates, although not as the primary source of traversability information. The models we construct are not based on range information, however, since this would prevent the extrapolation of the traversability prediction to regions where range is not available.

Howard et al. [8] presented a learning approach to determining terrain traversability based on fuzzy logic. A human expert was used to train a fuzzy terrain classifier based on terrain roughness and slope measures computed from stereo imagery. The fuzzy logic approach was also adopted by Shirkhodaie et al. [9], who applied a set of texture measures to windows of an image followed by a fuzzy classifier and region growing to locate traversable parts of the image.

Talukder and his colleagues [10] describe a system that attempts to classify terrain based on color and texture. Terrain is segmented using labels generated from a 3D obstacle detection algorithm. Each segment is described in terms of Gabor texture measures and color distributions. Based on color and texture, the segments are assigned to pre-existing classes. Each class is associated with an a priori traversability measure represented by a spring with known spring constant. We also make use of 3D obstacle detection in our work, but don't explicitly segment the data into regions. We model both background and obstacle classes using color and texture, but all models are created as the vehicle senses the world. Given that we have no prior knowledge of the type of terrain that may be encountered, it is usually not possible to pre-specify the classes. Similarly, the vehicle learns the traversability of the terrain by interacting with it, either by driving over it or generating a bumper hit.

## II. THE LEARNING ALGORITHM

The learning process takes input in the form of labeled pixels with associated  $(x, y, z)$  positions. The labels are provided on a pixel-by-pixel basis by an obstacle detection algorithm that works on stereo data [11]. Given the labels and color characteristics of the pixels, the learning algorithm constructs color and texture models of traversable and non-traversable regions and uses them for terrain classification. The approach to model building is to make use of the labeled color data to describe regions in the environment around the vehicle and to associate a cost of traversing each region with its description. The terrain models are learned using an unsupervised scheme that makes use of both geometric and appearance information.

In our algorithm an assumption is made that terrain regions that look similar will have similar traversability. The learning works as follows (see [12]). The system constructs a map of a 40 m by 40 m region of terrain surrounding the vehicle, with

map cells of size 0.2 m by 0.2 m and the vehicle in the center of the map. The map is always oriented with one axis pointing north and the other east. The map scrolls under the vehicle as the vehicle moves, and cells that scroll off the end of the map are forgotten. Cells that move onto the map are cleared and made ready for new information.

The model-building algorithm takes as input the color image, the associated and registered range data  $(x, y, z)$  points), and the labels (GROUND and OBSTACLE) generated by the obstacle detection algorithm. Also associated with these data is the location and pose of the vehicle when the data were collected. When new data are received, the vehicle location and pose information are used to scroll the map so that the vehicle occupies the center cell of the map.

Points are projected into cells based on their 3D positions. Each cell receives all points that fall within the square region in the world determined by the location of the cell, regardless of the height of the point above the ground. The cell to which the point projects accumulates information that summarizes the characteristics of all points seen by this cell. This includes color, texture, and contrast properties of the projected points, as well as the number of OBSTACLE and GROUND points that have projected into the cell. Color is represented by ratios R/G, G/B, and intensity. The intensity and color ratios are represented by 8-bin histograms stored in a normalized form so that they can be viewed as probabilities of the occurrence of each ratio. Texture and contrast are computed using Local Binary Patterns (LBP) [13]. These patterns represent the relationships between pixels in a 3x3 neighborhood in the image, and their values range from 0 to 255. The texture measure is represented by a histogram with 8 bins, also normalized. Contrast is represented by a single number ranging from 0 to 1.

When a cell accumulates enough points it is ready to be considered as a model. We determine the sample size by requiring 95% confidence that the sample represents the true distribution. In order to build a model, we also require that 95% of the points projected into a cell have the same label (OBSTACLE or GROUND). If a cell is the first to accumulate enough points, its values are copied to instantiate the first model. Models have exactly the same structure as cells, so this is trivial. If there are already defined models, the cell is matched to the existing models to see if it can be merged or if a new model must be created. Matching is done by computing a weighted sum of the squared difference of the elements of the model and the cell. Cells that are similar enough are merged into existing models; otherwise, new models are constructed.

At this stage, there is a set of models representing regions whose appearance in the color images is distinct (Fig 1). Our interest is not so much in the appearance of the models, but in the traversability of the regions associated with them. Traversability is computed from a count of the number of GROUND and OBSTACLE points that have been projected into each cell, and accumulated into the model. Models are given traversability values computed as  $N_{\text{OBSTACLE}} / (N_{\text{GROUND}} + N_{\text{OBSTACLE}})$ .

+  $N_{\text{OBSTACLE}}$ ). These models correspond to regions learned by example.

Learning by experience is used to modify the models. As the vehicle travels, it moves from cell to cell in the map. If it is able to traverse a cell that has an associated model, the traversability of that model is increased. If it hits an obstacle in a cell, the traversability is decreased.

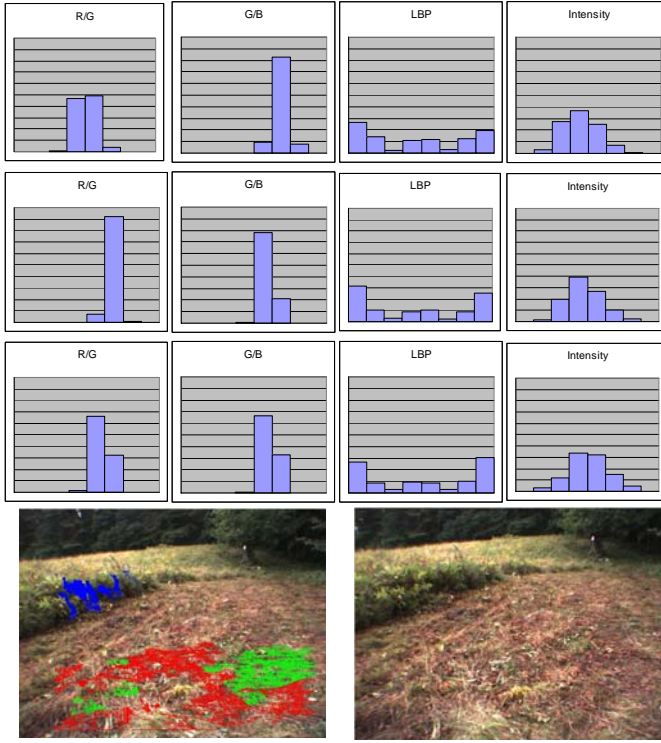


Fig 1. Examples of histograms used to construct models. Top row corresponds to the blue regions in the left image. Middle row corresponds to the green region. Bottom row corresponds to the red region. The blue region is not traversable, while the other two regions are traversable.

To classify a scene, only the color image is needed (no range data). A window is passed over the image and color, texture, and intensity histograms and a contrast value are computed as in model building. A comparison is made with the set of models, and the window is classified with the best matching model, if a sufficiently good match value is found. Regions that do not find good matches are left unclassified. Windows that match with models inherit the traversability measure associated with the model. In this way large portions of the image are classified (Fig 2).

The vehicle needs to know the locations of obstacle and ground regions, but has no stereo information during classification. To address this problem, the assumption is made that the ground is flat, i.e., that the pose of the vehicle defines a ground plane through the wheels. This allows windows that match with models to be mapped to 3D locations. Another assumption is that all obstacles (windows matching with models created from obstacle points) are

normal to the ground plane. This allows obstacle windows to be projected into the ground plane and thus to acquire 3D locations. Because of the ground plane assumption, the algorithm only processes the image from in front of the vehicle to a small distance above the horizon, to catch the obstacles but ignore the sky.

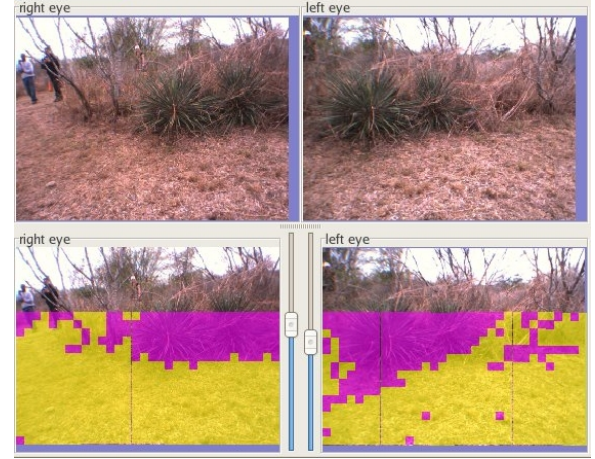


Fig 2. Top: Left and right eye views of a typical scene from Test 9. Bottom: Classification showing regions that are traversable in yellow, and not traversable in magenta.

### III. EVALUATING THE ALGORITHM

The entire LAGR system was tested over the course of a year by a separate Government team using a vehicle functionally identical to the vehicles on which the software is developed. Tests occurred about once a month. Developers sent their control software on flash memory cards to the test facility. The software was loaded onto a vehicle which was commanded to travel from a start waypoint to a goal waypoint through an obstacle-rich environment. The environment was not seen in advance by the development teams. The Government team measured the performance of the system on multiple runs. To demonstrate learning, performance was expected to improve from run to run as the systems became familiar with the course. While these tests gave a good indication of how learning improved the overall performance, they did not provide evaluations of individual learning algorithms.

Evaluating the algorithm described in this paper requires determining how well the learned models enable the system to classify the degree of traversability of the terrain around the vehicle. The evaluation makes use of ground truth generated by one or more human observers who use a graphical tool to generate ground truth points against which the learning algorithm is compared.

Data sets used for the evaluation consist of log files generated during the tests conducted by the Government team. Log files contain the sequence of images collected by the two pairs of stereo cameras on the LAGR vehicle and information from the other sensors, including the navigation (GPS and

INS) sensors and bumper sensors (physical and IR bumpers). The NIST LAGR system performs exactly the same when playing back a log file as it did when it first ran the course, so long as no changes are made in the algorithms. Therefore, logged data is a good source for performance testing.

The ground truth is collected by a human stepping sequentially through the log file, and classifying one or more points from each image. A graphical tool is used to display the image and randomly select a point (Fig. 3). The point is highlighted for the user, who selects one of the labels Ground (G), Obstacle (O), or Unknown (U). The tool then writes a record to a file containing the frame number, coordinates of the selected point, and the label provided by the user. Note that the Unknown label is used for points that are neither ground nor obstacle (such as sky) as well as points where the human truly cannot decide between ground and obstacle (such as at the base of an obstacle that merges smoothly with the ground). When ground truth collection is complete, the file is available for evaluating the performance of the learning algorithm (or any other algorithm that assigns traversability labels to regions).

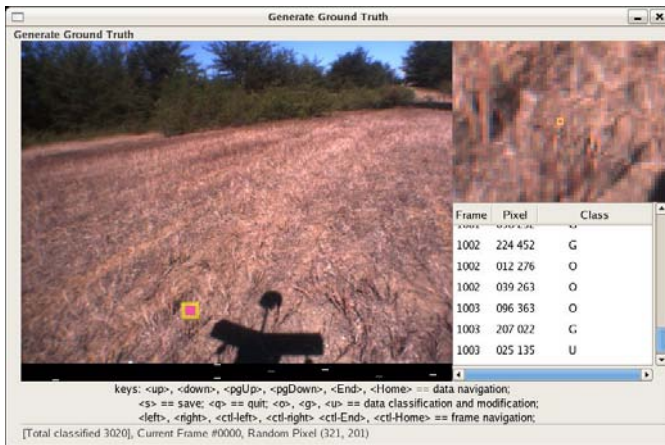


Fig. 3. The GUI for generating ground truth showing a frame from Test 7.

The learning algorithm reads the ground truth file and the log file. It processes the log file as it usually does when running on the vehicle. Each time it comes to an image frame for which ground truth is available, it classifies the points selected in the frame and writes out a file containing the ground truth it read in plus an entry giving the learned classification of the pixel in the ground truth file. When the entire log file has been processed, the output file contains an entry for each ground truth point that gives both the human's classification and the system's classification. Under the assumption that the human's classification is correct, an analysis can be conducted of the errors committed by the learning algorithm.

## IV RESULTS

The evaluation was applied to a number of examples taken from data gathered by the LAGR evaluation team at locations in Virginia and Texas. Results are shown for these examples and an overall evaluation is given of the performance of the algorithm across all the data sets.

In the evaluations, the learning system starts out with no models. This is how the system typically starts, at least for the first test run at each location. As it reads the log file and the ground truth data, the learning program both creates the models and classifies the ground truth points. This means that early in the sequence of images, only a small number of models are available for classification. As more of the terrain is seen, more models are constructed, and the range of regions that can be classified increases. The algorithm learns very fast, however, often creating the first few models from the first frame or two of data. Since the terrain doesn't usually change abruptly, classification performs well from the start, particularly for points close to the vehicle.

Four sets of ground truth data were created by three different people using the GUI in Fig. 3. The data were taken from log files of three different tests. Test 6 was conducted in September, 2005 in Fort Belvoir, VA. Test 7 was also conducted at Fort Belvoir, in October, 2005. The course was very different, however. Test 9 was conducted in San Antonio, TX at the Southwest Research Institute's Small Robot Testbed.

### A. Test 6

Test 6 included a run along a path through a slightly wooded area, ending in an open field. Two synthetic obstacles made out of orange plastic mesh were placed in the path of the vehicle (Fig. 4), with the goal being to learn that the first fence represented an obstacle and use that knowledge to avoid the second fence.



Fig. 4. A view of the first orange fence in Test 6.

The ground truth created for Test 6 consisted of approximately 3 points per frame, using the log file of the first test run. Because the human sometimes labeled a point as



Unknown, and because some of the points randomly selected for ground truth were in the sky, the actual number of usable points was closer to 2 per frame (there were 1,270 frames).

TABLE I shows a summary of the results of the evaluation. As can be seen, the algorithm labeled 87% of the points with the same class as the human. Of the incorrect labels, 30% arose from situations where the algorithm did not find a match with any model and labeled the points Unknown, 52% came from incorrectly labeling points as Obstacle instead of Ground, and 17% from labeling points as Ground instead of Obstacle.

TABLE I  
Results for Test 6

Test 6, 2513 Ground Truth Points			
No. Correct	No. Incorrect	% Correct	% Incorrect
2197	317	87.4%	12.6%
Error Distribution Across Label Types			
Not Classified (Unknown)	Obstacle instead of Ground	Ground instead of Obstacle	
30%	52%	17%	

#### B. Test 7

The course for Test 7 began in an open field. The straight-line path would put the vehicle in a position that required a long detour through dense bushes. Traveling to the right of the straight-line path led to an easy route to the goal. The Government team placed an artificial barrier in the path to make it difficult to choose the right hand direction the first time the course was seen (Fig. 5). The idea was that the vehicle would fight its way through the bushes on the first run before reaching the goal, but would learn to recognize the barrier and select the right hand route on subsequent test runs. In fact, this is what the NIST vehicle did.



Fig. 5. A view of the Test 7 course from the vehicle (on the wrong side of the barrier).

The ground truth for Test 7 was created from the log file of the first test run. Two different people generated ground truth files. One selected 1 point per frame, resulting in a usable count of 702 points, while the other selected 3 points per frame, resulting in a usable count of 2195 points, where usable points are determined as described above for Test 6. Having different selections of points for the same data set enabled us to see if there was any significant variation between people's selection of labels and also let us see if a smaller number of points was as effective as a larger one.

As can be seen in TABLE II and TABLE III, the results for both the small sample size and the large one are very similar, indicating that it is not necessary to label large numbers of points. What was surprising was that the distribution of the errors was different. For the smaller set, the percentage of errors due to the learning algorithm not being able to identify the class of the point was 46%, whereas the corresponding percentage for the larger set was 71%. In the tests we have done, the distributions of errors with different random sets of points has not shown any obvious pattern.

TABLE II  
Results for Test 7, User 1

Test 7, 702 Ground Truth Points			
No. Correct	No. Incorrect	% Correct	% Incorrect
592	110	84.5%	15.5%
Error Distribution Across Label Types			
Not Classified (Unknown)	Obstacle instead of Ground	Ground instead of Obstacle	
47%	34%	19%	

TABLE III  
Results for Test 7, User 2

Test 7, 2195 Ground Truth Points			
No. Correct	No. Incorrect	% Correct	% Incorrect
1884	312	85.8%	14.2%
Error Distribution Across Label Types			
Not Classified (Unknown)	Obstacle instead of Ground	Ground instead of Obstacle	
71%	4%	25%	

#### C. Test 9

Test 9 was conducted in the desert in December, 2005. The terrain was vegetated with both woodland and grassland features. The vegetation was dry, and there was not much color difference between the vegetation and the ground (Fig. 6). The course ran along a mowed path through the terrain, but

there were other paths crossing the desired path which did not provide a traversable route to the goal. The Government test team expected the vehicles to explore the side paths on the first run, but learn that they were not productive and follow the preferred path on later runs. This is what the NIST vehicle did.



Fig. 6. A view of the terrain in Test 9.

The ground truth for Test 9 was created from the log file of the first run, using a single point from each frame and a total of only 176 frames. There were a total of 290 points to be classified. As can be seen in TABLE IV, the system performed a little worse in this low-color environment, but still respectably.

TABLE IV  
Results for Test 9

Test 9, 290 Ground Truth Points			
No. Correct	No. Incorrect	% Correct	% Incorrect
232	58	80.3%	20.1%
Error Distribution Across Label Types			
Not Classified (Unknown)	Obstacle instead of Ground	Ground instead of Obstacle	
19%	21%	60%	

#### D. Cumulative Results

The results of all the performance evaluations are accumulated in TABLE V. As can be seen, 86% of the time the algorithm assigns similar labels to regions as do human observers.

TABLE V  
Cumulative Results

Tests 6, 7, and 9, 5701 Ground Truth Points	
Number of points classified	5701
Number correct	4905
Number incorrect	797
Percentage correct	86%
Percentage incorrect	14%

#### IV EVALUATING ALGORITHM PARAMETERS

Another way of using the ground truth data is to investigate the effects of the model parameters. We use five parameters, and here we discuss the effects of selecting subsets of these parameters. We explored using only color (no intensity or texture), using color plus intensity with no texture, and not using color. There are two color components, R/G and G/B. We did not explore removing only one of them. Nor did we look at the effects of contrast. Some of the results were surprising.

TABLE VI  
Effects on Classification of Changing Model Parameters

Test 7 Model Parameter Variation					
No Texture		No Color		Only Color	
% Correct	% Incorrect	% Correct	% Incorrect	% Correct	% Incorrect
83.52%	16.48%	53.26%	46.79%	86.25%	13.75%
Test 9 Model Parameter Variation					
No Texture		No Color		Only Color	
% Correct	% Incorrect	% Correct	% Incorrect	% Correct	% Incorrect
82.35%	17.99%	76.12%	24.22%	56.40%	43.94%

TABLE VI shows the classification success of the algorithm when it learns models with one or more features removed. It appears that removing texture has hardly any effect. The percentage of correct classifications for Test 7 goes down marginally (just over 2%), but the correct classification for Test 9 goes up (about 2%)! This is very surprising, since the data for Test 9 showed little color variation, so we assumed that the texture was providing most of the discrimination. It probably means that the texture measure we used is not suitable for this application (perhaps because it uses such a small neighborhood).

On the other hand, taking color out of the model features has a big impact, dropping the classification accuracy in Test 7 from about 86% to 53%. For Test 9 the accuracy also drops, but only from 80% to 76%. This is reasonable, since the data showed so little color variation.

Finally, if only color is used, the performance on Test 9 degrades considerably, from 80% to 56%. The performance on Test 7 actually goes up marginally, although probably not significantly. We can conclude that intensity plays a significant role in classification, especially in Test 9. Color is clearly important, but the use of the Local Binary Pattern operator is questionable.

## V. CONCLUSION

Knowing the traversability of terrain is very important to a robot that navigates off-road like those in the DARPA LAGR project. At NIST, we have developed several methods of learning traversability for use in the LAGR program. In this paper, we discussed our method of evaluating the performance of an algorithm that learns to classify terrain as either traversable or not traversable based on models it builds using color and texture features of the terrain.

The performance evaluation is not specific to the particular algorithm shown in this paper. Once a human has generated a set of ground truth points, they can be used to evaluate any classification algorithm. It is straightforward to modify the number of classes the user has available to classify the points, although too many classes may lead to a higher rate of human error in classifying the points. The evaluation was also applied to the stereo obstacle detection algorithm that provides the input for the learning algorithm and in some sense determines the best performance that can be expected of it. The results showed that the obstacle detection algorithm agreed with human classification 91% of the time.

The random nature in which the points to be classified are selected has the advantage of preventing any bias in the way that the image sequence is sampled. It has a problem, however, in that it is not possible to say anything about the way the errors are distributed in the images. There is a significant difference between errors that congregate at the boundaries of regions and those that appear throughout the image. Usually, errors close to boundaries are less of a concern since they amount to a disagreement about where the boundary actually occurs. Thus, two algorithms with the same performance in terms of correct classifications could differ greatly in their utility. The method used in this paper cannot provide a distinction based on error locations, but a quick scan of images such as Fig 2 gives a good idea of the error distribution.

It should be pointed out that the results shown in this paper do not take into account some postprocessing that is done in the algorithm after an image frame is classified but before the results are sent to the planner. This involves removing singleton blocks (16x16 windows of pixels) classified as one type that lie within a region of the opposite type (e.g., a single non-traversable block within a traversable region as can be seen in Fig 2). Usually such blocks are the result of incorrect classification so removing them improves the overall performance of the algorithm. In one of the tests (Test 10), however, the vehicles had to make their way through a set of thin posts randomly placed in a field. By removing singleton

blocks, the locations of some of the posts that had been correctly recognized by the algorithm as not traversable were lost.

It is very helpful to be able to use the performance evaluation to tune the algorithm by determining the useful features and their relative contributions to the final classification. Our evaluation showed that the texture operator was not performing effectively and that using intensity as a feature is beneficial. We plan to explore alternative texture measures based on multiresolution Gabor filters as in [10] to see if they perform better.

Overall, the results show that the algorithm for learning traversability works well, with a high degree of agreement between its classifications and those of a human observer. This provides confidence that the algorithm will enhance the performance of the LAGR control system as a whole.

## ACKNOWLEDGEMENT

The work described in this paper was conducted under a grant from the DARPA LAGR program. We are grateful for their support.

## REFERENCES

- [1] Jackel, L. Learning Applied to Ground Robots (LAGR). <http://www.darpa.mil/ipto/programs/lagr/>. 2005.
- [2] J. S. Albus, H-M Huang, E. Messina, K Murphy, M Juberts, A. Lacaze, S. Balakirsky, M. O Shneier, T. Hong, H. Scott, J. Horst, F Proctor, W. Shackleford, S. Szabo, and R. Finkelstein, "4D/RCS Version 2.0: A Reference Model Architecture for Unmanned Vehicle Systems," National Institute of Standards and Technology, Gaithersburg, MD, NISTIR 6912, 2002.
- [3] J. Albus, R. Bostelman, T. Chang, T. Hong, W. Shackleford, and M. Shneier, "Learning in a Hierarchical Control System: 4D/RCS in the DARPA LAGR Program," *Journal of Field Robotics, Special Issue on Learning in Unstructured Environments (in press)*, 2006.
- [4] Iwan Ulrich and Illah Nourbakhsh, "Appearance-Based Place Recognition for Topological Localization," IEEE International Conference on Robotics and Automation, 2000, pp. 1023-1029.
- [5] Iwan Ulrich and Illah Nourbakhsh, "Appearance-Based Obstacle Detection with Monocular Color Vision," *Proceedings of the AAAI National Conference on Artificial Intelligence*, 2000.
- [6] Ceryen Tan, Tsai Hong, Michael Shneier, and Tommy Chang, "Color Model-Based Real-Time Learning for Road Following," *Proceedings of the IEEE Intelligent*



Transportation Systems Conference, 2006.

- [7] Carl Wellington and Anthony Stentz, "Learning Predictions of the Load-Bearing Surface for Autonomous Rough-Terrain Navigation in Vegetation," International Conference on Field and Service Robotics, 2003, pp. 49-54.
- [8] Ayanna Howard, Edward Tunstel, Dean Edwards, and Alan Carlson, "Enhancing fuzzy robot navigation systems by mimicking human visual perception of natural terrain traversability," Joint 9th IFSA World Congress and 20th NAFIPS International Conference, 2001, pp. 7-12.
- [9] A. Shirkhodaie, R. Amrani, N. Chawla, and T. Vicks, "Traversable Terrain Modeling and Performance Measurement of Mobile Robots," Performance Metrics for Intelligent Systems, PerMIS '04, 2004.
- [10] A. Talukder, R. Manduchi, R. Castano, L. Matthies, A. Castano, and R. Hogg, "Autonomous Terrain Characterisation and Modelling for Dynamic Control of Unmanned Vehicles," IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2002, pp. 708-713.
- [11] T. Chang, T. Hong, S. Legowik, and M. Abrams, "Concealment and Obstacle Detection for Autonomous Driving," Proceedings of the Robotics & Applications Conference, Santa Barbara, CA, 1999, pp. 147-152.
- [12] Shneier.M., T. Chang, T. Hong, and W. Shackleford, "Learning Traversability Models for Autonomous Mobile Vehicles," *Autonomous Robots (submitted)*, 2006.
- [13] T. Ojala, M. Pietikainen, and D. Harwood, "A comparative study of texture measures with classification based on feature distributions," *Pattern Recognition*, vol. 29 1996, pp. 51-59.